Remarks

Applicants respectfully request that this Response After Final Action be admitted under 37 C.F.R. § 1.116.

Applicants submit that this Response presents claims in better form for consideration on appeal. Furthermore, applicants believe that consideration of this Response could lead to favorable action that would remove one or more issues for appeal.

No claims have been amended. No claims have been canceled. Therefore, claims 1-26 are now presented for examination.

Claims 1-26 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Alexander, Jr. et al. (U.S. Pub. No. 2003/0088689).Applicants submit that the present claims are patentable over Alexander.

Alexander discloses a streaming media accelerator that divides data received from sources into blocks and pre-calculates a checksum for each block. By "pre-calculating," it is meant that the checksum for a block of stream data received from streaming media sources is calculated in advance of the calculation of the checksums inserted in the headers of outbound data packets to be delivered to destinations. The pre-calculated checksums are used to calculate the checksums that are inserted in outbound packet headers. Because streams may be transmitted multiple times, calculating the checksums in advance greatly reduces the processing load on switch. See Alexander at paragraph [0029].

Claim 1 of the present application recites:

> A method comprising:
> partitioning data into segments of the data;
> storing in memory a set of checksums of the
> segments of the data, the set of checksums having a

first subset corresponding to the segments of data and a
second subset corresponding to the segments of data
shifted by a predetermined amount;

     selecting a portion of the data comprising at least
one of a subset of the segments of the data and at least
one part of at least one segment of the data; and

     determining a checksum of the portion of the data
based upon at least one of a checksum of the subset of
the segments of the data and a checksum of the at least
one part of the at least one segment of the data, the
checksum of the subset of the segments of the data
being based, at least in part, upon respective
checksums, read from the set of checksums stored in
the memory, of segments of the data comprised in the
subset of the segments of the data.

Applicants submit that there is no disclosure or suggestion in Alexander of a set

of checksums having a first subset corresponding to the segments of data and a second

subset corresponding to the segments of data shifted by a predetermined amount. In fact,

at one instance the Final Office Action admits that Alexander does not disclose or suggest

such a feature. See Final Office Action at page 3, ll. 9-12. However, in the same

paragraph the Final Office Action asserts that Alexander does disclose the feature in

paragraphs [0034] – [0038]. See Final Office Action at page 3, ll. 16-19.

Applicants agree with the first assertion that Alexander does not disclose a set of

checksums having a first subset corresponding to the segments of data and a second

subset corresponding to the segments of data shifted by a predetermined amount. The

passage of Alexander relied upon discloses:

[0034] FIG. 4 is a block diagram illustrating an exemplary
method and systems for pre-calculating checksums
according to an embodiment of the present invention. In
FIG. 4, block 400 represents a stream of data received from
a multimedia source. The data is divided into blocks 402 of
a predetermined size. For example, blocks 402 may each be
128 bytes in size. Blocks 402 may be further subdivided
into smaller blocks for checksum calculations. For
example, since the checksum field in both the TCP and
UDP headers is 16 bits, blocks 402 may each be subdivided
into 16-bit units. Once blocks 402 have been subdivided

into 16-bit units, a checksum is computed by adding the 16-bit units in each block, for example, using one's complement addition. Accordingly, the present invention may include a checksum pre-calculator 403 for pre-calculating checksums. Checksum pre-calculator 403 may be implemented in software executing on one or more of the processors resident on streaming media accelerator 102.

[0035] As stated above, one method for pre-computing the checksum in each block, which decreases the time required to compute the checksum for each stream, is computing running checksums. The running checksum for a given block is the sum of the checksum for that block and the running checksum for the previous block. In the case where there are no previous blocks, for example, in FIG. 4, the running checksum for point 404 in the data stream is calculated based on the bits in block 0 only. The checksum for point 406 in the data stream is the sum of the checksum for block 1 and the running checksum for block 0. Similarly, the checksum stored for point 408 in the data stream is the sum of the checksum for block 2 and the running checksum for block 1. The checksum stored for point 410 at the end of block n in the data stream is then the sum of the checksum for block n and the running checksum for block n-1.

[0036] As the running checksum values are calculated, they are preferably stored in memory for use in calculating the checksums to be included in outbound data packets. As stated above, in streaming media accelerator 102, the checksums for each stream may be stored in memory device 206 accessible by a table lookup unit 214. FIG. 5 is a block diagram illustrating in further detail the calculation of a data checksum performed based on running checksums according to an embodiment of the invention. In FIG. 5, a packet checksum calculator 500, which may be implemented in software executing on channel processors 200 illustrated in FIG. 2 or any other suitable processor, receives instructions to send data blocks 1 through 3 of stream 1. Packet checksum calculator 500 informs table lookup unit 214 that the running checksums for blocks 0 and 3 are required. Table lookup unit 214 performs a lookup in table 502 and extracts the running checksums for the ends of blocks 0 and 3 from the checksum data stored for stream 1. Table lookup unit 214 outputs this data to packet checksum calculator 500 Packet checksum

calculator 500 computes the difference of the checksums stored for blocks 0 and 3 and outputs a differential checksum for blocks 1 through 3.

[0037] Computing a differential checksum may be accomplished using one's complement arithmetic by adding the checksum from block 3 to the complement of the checksum of block 0. For example, if the running checksum stored for block 3 is 1010101100110111 and the checksum stored for block 0 is 1110011110110110, the differential checksum may be calculated as follows: 1 Differential Checksum Calculation _ 1010101100110111 + 0001100001001001 _ 1100001110000000 (1)

[0038] In equation 1, the top addend is the running checksum for block 3. The bottom addend is the complement of the running checksum for block 0. The sum is the difference in the checksums between blocks 3 and 0. Thus, it is apparent from equation 1 above that a differential checksum calculation for a large number of data blocks involves simple one's complement addition of only two checksums, which greatly reduces the calculation over conventional calculation methods that compute checksums, 16 bits at a time, for each portion of data to be sent. As stated above, an alternate method for pre-calculating checksums is to calculate the checksums for each block and store the pre-calculated checksums for each block in memory. Referring back to FIG. 4, the checksum for each block 402 would simply be the one's complement sum of each 16-bit sub-block of each block. In order to calculate a checksum for an outgoing data packet using these pre-calculated checksums, packet checksum calculator 500 illustrated in FIG. 5 would simply sum the checksums for each block to be included in an outgoing data packet. For example, if it is desirable to send blocks 0 through 3 to a destination, the checksum for blocks 0 through 3 would be calculated by summing the pre-calculated stored checksum values for block 0, block 1, block 2, and block 3. This checksum calculation involves more computation when data is sent, compared to the first method above. However, because the pre-calculated checksum values are reused each time a data stream is requested by a destination, significant time and processing savings are achieved over conventional TCP and UDP checksum calculation routines.

Applicants submit that nowhere in the above-passage relied upon by the Final office action is there disclosed or suggested a second subset corresponding to segments of data shifted by a predetermined amount. Therefore, Alexander does not teach or suggest all the limitations of claim 1, as required by 35 U.S.C. §103(a).

Claims 2-7 depend from claim 1 and include additional features. Therefore, claims 2-7 are also patentable over Alexander.

Claim 8 recites:

> An apparatus comprising:
> circuitry that is capable of:
> partitioning data into segments of the data;
> storing in memory a set of checksums of the segments of the data, the set of checksums having a first subset corresponding to the segments of data and a second subset corresponding to the segments of data shifted by a predetermined amount;
> selecting a portion of the data comprising at least one of a subset of the segments of the data and at least one part of at least one segment of the data; and
> determining a checksum of the portion of the data based upon at least one of a checksum of the subset of the segments of the data and a checksum of the at least one part of the at least one segment of the data, the checksum of the subset of the segments of the data being based, at least in part, upon respective checksums, read from the set of checksums stored in the memory, of segments of the data comprised in the subset of the segments of the data.

For the reasons described above with respect to claim 1, claim 8 is also patentable over Alexander. Since dependent claims 9-14 depend from claim 8, and include additional features, claims 9-14 are also patentable over Alexander.

Claim 15 recites:

> An article comprising:

a storage medium that stores instructions that when executed by a machine result in the following:

partitioning data into segments of the data;

<u>storing in memory a set of checksums of the segments of the data, the set of checksums having a first subset corresponding to the segments of data and a second subset corresponding to the segments of data shifted by a predetermined amount;</u>

selecting a portion of the data comprising at least one of a subset of the segments of the data and at least one part of at least one segment of the data; and

determining a checksum of the portion of the data based upon at least one of a checksum of the subset of the segments of the data and a checksum of the at least one part of the at least one segment of the data, the checksum of the subset of the segments of the data being based, at least in part, upon respective checksums, read from the set of checksums stored in the memory, of segments of the data comprised in the subset of the segments of the data.

For the reasons described above with respect to claim 1, claim 15 is also patentable over Alexander. Because dependent claims 16-21 depend from claim 15, and include additional features, claims 16-21 are also patentable over Alexander.

Claim 22 recites:

A system comprising:

a circuit board that includes a circuit card slot; and

a circuit card that is capable of being coupled to the circuit board via the circuit card slot, the circuit card including circuitry that is capable of:

<u>storing in memory a set of checksums of the segments of the data, the set of checksums having a first subset corresponding to the segments of data and a second subset corresponding to the segments of data shifted by a predetermined amount;</u>

selecting a portion of the data comprising at least one of a subset of the segments of the data and at least one part of at least one segment of the data; and

determining a checksum of the portion of the data based upon at least one of a checksum of the subset of the segments of the data and a checksum of the at least one part of the at least one segment of the data, the

checksum of the subset of the segments of the data
being based, at least in part, upon respective
checksums, read from the set of checksums stored in
the memory, of segments of the data comprised in the
subset of the segments of the data.

Thus, for the reasons described above with respect to claim 1, claim 22 is also patentable over Alexander. Since dependent claims 23-26 depend from claim 22, and include additional features, claims 23-26 are also patentable over Alexander.

Applicants respectfully submit that the rejections have been overcome, and that the claims are in condition for allowance. Accordingly, applicant respectfully requests the rejections be withdrawn and the claims be allowed.

The Examiner is requested to call the undersigned at (303) 740-1980 if there remains any issue with allowance of the case.

Please charge any shortage to our Deposit Account No. 02-2666.

Respectfully submitted,
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: June 12, 2006

Mark L. Watson
Reg. No. 46,322

12400 Wilshire Boulevard
7th Floor
Los Angeles, California 90025-1026
(303) 740-1980